REMARKS

The present amendment is in response to the Office Action dated November 1, 2007. Claims 1-17 are now present in this case. Claims 1, 2, 8, and 12-17 are amended.

Claims 12-17 are objected to because of the following informalities: These claims all recite, or refer back to, "A computer readable media ... ", which is grammatically incorrect. Pursuant to the Examiner's suggestion, claim 12 and all claims depending from claim 12 have been corrected to read "The computer readable medium ... ".

Claims 8-11 stand rejected under 35 U.S.C. § 101 as directed to non-statutory subject matter. The applicants have amended the claims in accordance with the Examiner's suggestion and are now believed to be properly directed to statutory subject matter. Accordingly, the applicants respectfully request that the Examiner withdraw the rejections of claims 8-11 under 35 U.S.C. § 101.

Claims 1-7 stand rejected under 35 U.S.C. § 102(b) as being anticipated by U.S. Patent No. 4,744,048 to Blanset et al. The applicants respectfully traverse this rejection and request reconsideration. Blanset et al. is directed to a solution for what is now outdated microprocessor technology. At the time Blanset was written, MS-DOS operated in a non-protected mode while UNIX operating systems operated in a protected mode. The problem addressed by Blanset et al. is that switching context from one operating system to another resulted in the potential problem that an MS-DOS application program could over-write the display buffer used by the UNIX operating system. Because UNIX operates in a protected mode, access to the display buffer can be controlled by the operating system. In contrast, MS-DOS operated in an unprotected mode thus giving application programs unfettered access to the display buffer. This caused potential conflicts because the MS-DOS application program could over-write the display buffer utilized by the UNIX operating system. (See Abstract.)

To solve this problem, Blanset et al. describes a technique for creating an alternate buffer for use by MS-DOS. Blanset et al. propose a hardware solution in which a hardware component monitors addresses generated by the computer and

selectively inhibits control pulses to prevent the MS-DOS system from writing to the display buffer utilized by the UNIX system. (See Abstract.) Thus, data generated by an MS-DOS application is shunted to the alternate display buffer while application program data generated on the UNIX operating system is directed to the protected display buffer.

It is important to note that Blanset et al. only allows the display of data from only one operating system at a time. Specifically, Blanset et al. discloses a context switching mechanism whereby a user can select, via a keyboard operation, to display the image created from the UNIX operating system or the image generated from the MS-DOS operating system. (See Abstract, and col. 8, lines 24-55.) Thus, Blanset et al. disclose a technique for allowing two independent operating systems to execute on a single microprocessor and display the data from either the MS-DOS operating system or the UNIX operating system. However, in no case can the system of Blanset display the data from both the MS-DOS application and the UNIX application at the same time.

In contrast to Blanset, the method of claim 1 is not intended to operate on two independent operating systems. Rather, claim 1 recites a method that allows data from an authorized data source to be displayed on a specified region of the video display system and prevents an unauthorized application running on the native operating system from over-writing that display region. However, it is important to note that the method of claim 1 does not recite two operating systems, as does Blanset. In addition, the authorized application, in fact, utilizes resources from the native operating system to write data to the display area. Specifically, claim 1 recites *inter alia* "upon receiving an indication from the authorized display source to write the image within the area defined by the associated display region mask, utilizing resources from the native system to write the image onto the display area, such that the output from the unauthorized source is not displayed within the area defined by the associated display region mask." Support for this concept may be found in the application as originally filed at pages 34-36. As detailed in that section and illustrated in Figure 32, the inventive method determines whether applications that are unauthorized are attempting to write to the defined display area of the video display system and intercepts those calls. However, the output is still delivered to the display driver. As those skilled in the art will appreciate, the display driver is typically written to provide a data interface between the

operating system and the display hardware. Thus, the display driver may be considered to be part of the native operating system since it is uniquely developed for each particular operating system. Blanset does not teach or suggest any technique by which the authorized data utilizes the native operating system resources, as recited in claim 1. Accordingly, claim 1 is clearly allowable over Blanset et al.

Claim 2 is directed to a method for preventing a first application from over-writing data displayed by a second application on a video display system. Claim 2 recited *inter alia* "modifying a portion of the received data intended for the display area defined by the display region mask to prevent the date from the first application from being displayed in the display area defined by the display region mask" as well as "transferring the data, including the modified portion, to a display driver associated with the video display system wherein the video display system displays data from the first application except in the display area defined by the display region mask and simultaneously displays data from the second application in the display area defined by the display region mask."

As discussed above, Blanset discloses a technique for protecting a video buffer from being overwritten by an MS-DOS program operating in an unprotected mode. However, the system of Blanset does not allow the simultaneous display of data from the two operating systems. The user can select data from the UNIX operating system for display or, by the appropriate context switch using the keyboard operation, can change context to the MS-DOS application and thereby display only the MS-DOS data. Blanset does not teach or suggest any method by which data from the two applications is simultaneously displayed on the display where the data from the first application cannot be written in the display area defined by the display area region mask. Accordingly, claim 2 is clearly allowable over Blanset et al.

Claims 3-7 are also allowable in view of the fact that they depend from claim 2, and further in view of the recitation in each of those claims. For example, claim 3 recites a method in which the modification of the data is performed by a display filter positioned intermediate the graphic display interface and the vide display driver. Those skilled in the art will appreciate that these are software components that function within the environs of the operating system. In contrast, Blanset discloses a hardware device

that monitors the hardware address bus and simply prevents writing of data to certain selected hardware addresses. This does not suggest the method of claim 3 in which the software device is positioned intermediate the graphic display interface and the vide display driver to function as a software data filter.

Claims 8-17 stand rejected under 35 U.S.C. § 103(a) as unpatentable by NN96057 in view of Blanset et al.. The applicants respectfully traverse this rejection and request reconsideration. NN96057 discloses a technique for coordinating multiple graphical user interfaces (GUIs) in a single operating system. The solution proposed by NN96057 is to enable only one GUI at a time. If an application attempts to write to a GUI that is currently disabled, that data is accumulated (i.e., stored) for later use. (See page 1). When the GUI is subsequently enabled, a return code indicates that the GUI is enabled and any accumulated data is now redrawn. (See page 1). Thus, NN96057 merely describes a system where GUIs are enabled or disabled. While a GUI is disabled, any data is stored for future use. When the GUI is enabled, the data is written.

The Office Action states that "NN96057 does not disclose a system that creates a display region mask that defines a mask display area of the video display system." (See Office Action, page 8). The applicants agree with this assessment. However, on page 7 the Office Action states that NN96057 does disclose a programming interface "to provide a routine to associate the generated display region mask with the second application." The applicants disagree with this assertion. If NN96057 does not disclose a system that creates a display region mask, is cannot be held to disclose associating the generated display region mask with the second application. The addition of Blanset to NN96057 does not overcome this serious deficiency. As discussed above, Blanset describes a hardware solution that physically monitors address lines on the address bus of the microprocessor and prevents the generation of control signals to store data in selected memory locations that define the display buffer.

In sharp contrast, the system recited in claim 8 recites *inter alia* "a graphics display device interface configured to receive graphic display interface (GDI) calls from a processor executing the first and second applications." The hardware

solution of Blanset focuses solely on a particular address space and prevents writing from the MS-DOS program to specific hardware address locations. Thus, Blanset has no knowledge of the actual data being sent to those address locations. Similarly, NN96057 has no knowledge of the content of data being transmitted by any particular GUI. NN96057 merely checks to determine whether the GUI is enabled or disabled. If the GUI is enabled, the data is processed regardless of any location on the display screen. If the particular GUI is disabled, nothing is written to the display screen. Thus, the combination of Blanset and NN96057 do not suggest any technique by which GDI calls are intercepted and analyzed to determine whether the data within the GDI is specifying transmission of data to the masked display area nor do the combination of references suggest clipping a portion of the received data to prevent that data from being written to the masked display area. Thus, claim 8 is clearly allowable over the combination of NN96057 and Blanset et al. Claims 9-11 are also allowable in view of the fact that they depend from claim 8 and further in view of the recitation in each of those claims.

Claim 12 is a computer readable medium claim for controlling the processor to thereby prevent a first application from over-writing data displayed by a second application on a video display system. Claim 12 recites *inter alia* "clipping a portion of the received data intended for the display area defined by the display region mask to prevent the data from the first application from being displayed in the display area defined by the display region mask wherein the video display system displays data from the first application except in the display area defined by the display region mask and simultaneously displays data from the second application in the display area defined by the display region mask." As discussed above with respect to claim 2, Blanset does not teach or suggest any mechanism by which the two application programs may simultaneously display data on the display screen. Blanset is directed to two totally independent operating systems whereby only one of the operating systems is actively displaying data. Blanset discloses a keyboard operation whereby the user can change context between the UNIX operating system and MS-DOS to thereby change the data displayed on the display system. However, at no time does Blanset disclose a technique for displaying data from both systems simultaneously. NN96057 does not

overcome this serious deficiency. NN96057 discloses a display controller that allows one GUI to display data at a time. If data is provided for a disabled GUI, it is accumulated until a subsequent time when that GUI is enabled. At that subsequent time, the accumulated data is displayed. NN96057 does not teach or suggest any mechanism by which the data from two applications are displayed simultaneously where data is received from a first application from a graphic device interface associated with the native operating system and clipped in order to prevent the display of data from the first application within the display area defined by the display region mask. Accordingly, claim 12 is allowable over the combination of NN96057 and Blanset. Claims 13-17 are also allowable in view of the fact that they depend from claim 12, and further in view of the recitation in each of those claims.

In view of the above amendments and remarks, reconsideration of the subject application and its allowance are kindly requested. The applicant has made a good faith effort to place all claims in condition for allowance. If questions remain regarding the present application, the Examiner is invited to contact the undersigned at (206) 757-8029.

Respectfully submitted,

Davis Wright Tremaine LLP

/Michael J. Donohue, Reg. #35859/

Michael J. Donohue

MJD:mn

1201 Third Avenue
Suite 2200
Seattle, Washington 98101
Phone: (206) 757-8029
Fax: (206) 757-7029

2286550_1.DOC